1987-03

The Archipelagian Approach to DSS prototyping: an empirical study

Albers, Stephen George

Monterey, California: U.S. Naval Postgraduate School

http://hdl.handle.net/10945/22644

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE ARCHIPELAGIAN APPROACH TO DSS PROTOTYPING: AN EMPIRICAL STUDY

by

Stephen George Albers

March 1987

Thesis Advisor: T. R. Sivasankaran

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)		10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) THE ARCHIPELAGIAN APPROACH TO DSS PROTOTYPING: AN EMPIRICAL STUDY

12 PERSONAL AUTHOR(S) Albers, Stephen George

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987 March	45

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Decision Support System (DSS); DSS prototyping; Archipelagian Approach

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The Archipelagian Approach, a proposed methodology for designing and implementing Decision Support Systems (DSS), attempts to integrate modular design and adaptive design. The approach is based on decomposing the proposed system's tasks into structured and nonstructured modules, evaluating the difficulty of implementing each module, and utilizing the estimated difficulty and the priority of each module to determine the best development sequence. The feasibility of making reliable and accurate predictions of implementation difficulty, a key requisite, was previously not verified. This thesis presents a discussion of the Archipelagian Approach and an empirical study of factors that potentially could be used to predict implementation difficulty. The study concludes that five of the eight factors considered exhibit sufficient reliability and validity as predictors to confirm the viability of the approach.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. T.R. Sivasankaran	22b TELEPHONE (Include Area Code) (408) 646-2637	22c OFFICE SYMBOL Code 54Sj

83 APR edition may be used until exhausted
All other editions are obsolete

The Archipelagian Approach to DSS Prototyping: An Empirical Study

by

Stephen George Albers
Lieutenant, United States Navy
B.G.S., University of Michigan, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1987

ABSTRACT

The Archipelagian Approach, a proposed methodology for designing and implementing Decision Support Systems (DSS), attempts to integrate modular design and adaptive design. The approach is based on decomposing the proposed system's tasks into structured and nonstructured modules, evaluating the difficulty of implementing each module, and utilizing the estimated difficulty and the priority of each module to determine the best development sequence. The feasibility of making reliable and accurate predictions of implementation difficulty, a key requisite, was previously not verified. This thesis presents a discussion of the Archipelagian Approach and an empirical study of factors that potentially could be used to predict implementation difficulty. The study concludes that five of the eight factors considered exhibit sufficient reliability and validity as predictors to confirm the viability of the approach.

TABLE OF CONTENTS

LIST OF TABLES

I. INTRODUCTION

A. GENERAL BACKGROUND

The Archipelagian Approach, a proposed methodology for designing and implementing Decision Support Systems (DSS), attempts to integrate and capture the advantages of both modular design and adaptive design. A key step requires the DSS builder to accurately estimate module accomplishability, a prediction of the difficulty of implementing each module in the planned project. If no reliable, valid prediction measure is feasible, then the proposed methodology becomes merely an academic exercise without a functional application. This thesis presents a discussion of the Archipelagian Approach, and an empirical evaluation of potential factors that could be utilized to estimate module accomplishability.

B. OBJECTIVE

The objective of this thesis is to evaluate the viability of the Archipelagian Approach requirement to predict module accomplishability. The study identifies possible factors or variables that could serve as predictors of accomplishability, and assesses the reliability and validity of the estimates made when these factors are utilized to evaluate sample modules.

C. RESEARCH QUESTIONS

Four main questions are addressed. (1) What factors should be used to estimate accomplishability? (2) What is the inter-rater reliability for each factor? (3) How valid are the implementation feasibility predictions made using each factor? (4) What conclusions can be drawn about the viability of the Archipelagian methodology?

D. SCOPE AND LIMITATIONS

In addition to the implementation difficulty prediction, or Accomplishability Factor (AF), the Archipelagian Approach utilizes an Imperative Factor (IF) to express the priority associated with each module, and a Development Priority Factor (DPF) to determine module development sequence. The IF and DPF are explained in the background discussion, but not addressed in the empirical portion of the study.

The Archipelagian Approach is intended for use by Decision Support System (DSS) builders. Responses from practitioners would have been preferred for evaluating the reliability and validity of potential accomplishability measures. However, to facilitate collecting data, a survey of graduate students in the fifth quarter of the Computer Systems Management (367) curriculum at the Naval Postgraduate School was used.

E. LITERATURE REVIEW AND METHODOLOGY

This research is based on issues raised in a paper by T. X. Bui and T. R. Sivasankaran of the Naval Postgraduate School faculty titled "Integrating Modular Design with Adaptive Design in DSS Prototyping: An Archipelagian Approach" in which the concept is proposed [Ref. 1]. A summary of the approach appears in Chapter Two of this thesis.

The methodology for conducting the study included four steps. (1) Review of literature to identify factors that DSS researchers postulate could affect accomplishability. (2) Design of a questionnaire containing narrative descriptions of modules for respondents to evaluate using the selected factors. (3) Administration of the questionnaire to a group of NPS graduate students. (4) Statistical analysis of the results using Minitab Release 5.1 running under VM/CMS on an IBM 3Ø33 computer.

F. SUMMARY OF FINDINGS

Eight possible module implementation difficulty predictors or factors were selected for the study. The factors were Task Complexity, Task Programmability, Task Structure, Module Size, Tool Availability, Value Judgement, Task Analyzability, and Completion Time. The first five factors listed exhibited significantly higher inter-rater reliability. The validity of the factors as predictors of accomplishability using estimates made by individual raters was disappointingly low; however, the results using the group means were highly accurate, demonstrating that prediction of accomplishability is practical if aggregate judgements on the factors are used. The high correlation coefficients among the high-reliability factors imply that they are largely redundant. The study did not undertake a factor analysis to determine which of the factors should be eliminated.

G. ORGANIZATION OF STUDY

Chapter Two presents the theoretical background for the study, including discussions of modular design, adaptive design, and the Archipelagian Approach. Chapter Three describes the study methodology, focusing on the construction of the questionnaire, and summarizes the collected data. Chapter Four contains the analysis results and possible interpretations. The closing chapter presents conclusions, recommendations, and questions for further research.

II. BACKGROUND AND THEORETICAL FRAMEWORK

A. MODULAR DESIGN

Modular or structured systems design is a disciplined methodology for computer system design that evolved in an attempt to avoid the high cost and poor maintainability associated with earlier software development methods. With modular design, large complex systems are partitioned into simple, independent blackbox modules organized into hierarchies suitable for computer implementation. The methodology includes graphic tools for easy communication of specifications and design results, a set of strategies for developing design solutions, and a set of criteria for evaluating the quality of the resulting design solution. [Ref. 2]

Several advantages can be realized through the use of modular techniques. First, complex systems are more easily understood when partitioned into simple modules. Second, development is more rapid because modules can be coded and tested in parallel and reused in other projects. Third, the graphic tools of modular design provide good system documentation. Fourth, modular systems are more reliable, easier to modify, and less expensive to maintain. [Ref. 3]

B. ADAPTIVE DESIGN

Despite the advantages, modular design has generally not been applied to the development of Decision Support Systems. The poorly-structured nature of DSS tasks, and the evolutionary nature of the DSS environment, preclude the complete, one-step specification of functional requirements for the system in advance [Ref. 4]. Instead, researchers advocate the adaptive design strategy. Adaptive design is an iterative technique in which the final system emerges through a series of prototypes. The initial prototype, produced quickly and on a small scale, represents computer-based support of a limited subproblem. Through interaction with the initial system, users develop new perceptions and insights which stimulate the need for new functions; these new requirements are incorporated into the next generation prototype by the builders. This interaction between users and builders continues until a satisfactory final system is completed. [Ref. 5]

Adaptive design provides the flexibility necessary to approach the automation of poorly-structured tasks. However, the strategy treats the entire system as poorly-structured, leading to unnecessary and costly interactions between the users and builders over well-defined functions [Ref. 6]. In addition, prototyping ignores the potential benefits of modular design.

Either method can lead the DSS development team to waste time, effort, and resources on a project that is ultimately abandoned because some key feature cannot be implemented.

C. THE ARCHIPELAGIAN APPROACH

The Archipelagian Approach to DSS prototyping attempts to secure the advantages of the modular design method while maintaining the flexibility of the adaptive design strategy. The approach consists of four basic steps [Ref. 1].

1. Step One

Decompose the proposed DSS into as many functional subsystems as possible, and decompose each subsystem into its component modules. This results in dividing a complex poorly-structured problem into "islands" of both structured and ill-structured subproblems (and provides the inspiration for the name "archipelagian").

2. Step Two

Compute an Accomplishability Factor (AF) for each module. In the initial paper, the AF is conceived as a function of Perceived Task Structure and Tool Availability, and is expressed on a scale from zero (very low) to one (very high). Modules with a very high AF would be relatively easy to implement and are suitable for structured design and implementation techniques, while modules with a lower AF entail more risk and probably require implementation through the adaptive design methodology.

3. Step Three

Specify an Imperative Factor (IF) for each module. This factor allows the incorporation of user priorities and implementation sequence constraints into the development strategy. Modules representing functions that the users desire the most, or that must be completed as a prerequisite to building some other modules, will have an IF close to zero. Modules that can wait are assigned an IF closer to one.

4. Step Four

Compute a Development Priority Factor (DPF) for each module to aid the DSS Builder in determining a module development sequence that will progressively reduce project risk. The DPF is the product of the AF and the IF. Since the more risky modules will have a low DPF as a result of their low AF, implementing modules in order from low to high DPF will minimize wasted effort if the project is cancelled because of inability to implement a risky module.

The key to the Archipelagian Approach is the second step, computing the Accomplishability Factor. Without a valid, reliable method of predicting the likelihood of successfully implementing each module, the approach cannot be applied. The remainder of this study addresses the accomplishability prediction issue.

III. METHODOLOGY AND DATA

A. QUESTIONNAIRE CONSTRUCTION

The survey questionnaire was developed through a three-step process. (1) Potential variables or factors that could affect accomplishability were selected for study. (2) A precise definition and rating scale was drafted for each selected factor. (3) Narrative descriptions of sample program modules were prepared for respondents to evaluate using the specified factor definitions and rating scales. The complete questionnaire is included as the Appendix to this thesis.

Potential factors were evaluated against three criteria. First, there had to be at least an intuitive sense that the factor was likely to affect the ability to implement program modules. Second, the factor had to have potential variability across different modules in the same project, not only across different projects or development teams. Third, the factor had to be capable of expression in the form of a scale or standards that individuals could use in making judgements on the modules. The factors selected for inclusion in the questionnaire are listed in Table 1 below, along with the abbreviations that will be used in the data summary and analysis portions of the report. In addition to the eight factors, Estimated Accomplishability was incorporated to represent a summary judgement.

POTENTIAL FACTORS

TABLE 1

Factor	Abbreviation
Task Complexity	cmplx
Task Programmability	prog
Task Structure	struc
Task Analyzability	analy
Value Judgement	value
Tool Availability	tool
Module Size	size
Completion Time	time
Estimated Accomplishability	est.acc

The definition prepared for each factor emphasized its applicability to the implementation of tasks at the module level. With the exception of Completion Time, each factor was assigned a rating scale between zero and one, with five possible values for respondents to choose. A description for each value provided a standard that the sample modules could be compared against. Respondents were asked to estimate Completion Time for sample modules directly in man-hours. The complete set of definitions and rating scale descriptions is listed in the Appendix.

The twelve sample modules for the questionnaire were selected with the goal of covering a variety of situations and implementation difficulties. To allow estimation of the actual accomplishability, modules from existing DSS projects were utilized. Modules one [Ref. 7], two through five [Ref. 8], and nine through twelve [Ref. 9] originated in previous Naval Postgraduate School thesis projects; modules seven and eight were inspired by commercial products [Ref. 10]. The description for module six was deliberately drafted to represent a task that is currently impossible for a computer program. The actual accomplishability assigned to each module represents a judgement by this researcher based on the degree to which the module meets its stated purpose, and on the degree of implementation difficulty reported by the module's authors. The values for actual accomplishability appear in the "actual" column of Table 2 in the Collected Data section of this report.

B. SAMPLE SIZE AND DEMOGRAPHICS

The questionnaire was administered to 47 students at the Naval Postgraduate School on December 10 and 11, 1986. Two of the returned forms were incomplete, leaving 45 usable responses. Of the 45 respondents, 39 were students in the fifth quarter of the Computer Systems Management (367) curriculum, three were in the third quarter of the 367 curriculum, and three were from the Telecommunications System Management (620) curriculum. Only eight of the respondents reported any experience with computer software design or development outside of their course work.

C. COLLECTED DATA

The table on the following page lists the sample mean (m) and sample standard deviation (s) of the values selected by all respondents for each factor in rating each module. Completion Time is expressed in hundreds of man-hours; all other factors are in terms of the zero to one scale.

COLLECTED DATA SUMMARY

TABLE 2

module	cmplx		prog		struc		analy		value	
	m	s	m	s	m	s	m	s	m	s
1	.64	.22	.83	.14	.79	.19	.47	.26	.73	.25
2	.52	.27	.69	.20	.70	.22	.59	.22	.78	.19
3	.52	.21	.69	.16	.70	.17	.51	.24	.73	.23
4	.35	.15	.54	.17	.52	.18	.54	.24	.70	.20
5	.27	.21	.39	.21	.38	.20	.46	.27	.49	.27
6	.27	.20	.39	.21	.40	.23	.52	.27	.53	.27
7	.64	.20	.73	.15	.72	.20	.60	.22	.71	.21
8	.53	.19	.66	.15	.64	.20	.59	.22	.66	.23
9	.74	.20	.82	.14	.82	.17	.63	.30	.71	.24
10	.48	.22	.63	.23	.56	.24	.61	.21	.59	.27
11	.62	.22	.70	.17	.68	.20	.53	.28	.53	.26
12	.96	.12	.96	.12	.96	.10	.55	.40	.70	.28

module	tool		size		time		est.acc		actual
	m	s	m	s	m	s	m	s	
1	.92	.13	.62	.25	1.6	2.9	.79	.16	0.75
2	.71	.21	.46	.26	6.5	14.2	.69	.22	0.75
3	.74	.23	.42	.17	4.7	10.7	.66	.15	0.75
4	.58	.24	.29	.18	10.6	16.1	.51	.18	0.50
5	.39	.24	.15	.16	85.4	218	.34	.17	0.25
6	.45	.24	.19	.16	76.0	212	.38	.20	0.00
7	.80	.18	.39	.18	7.1	11.9	.69	.19	0.50
8	.75	.21	.38	.18	11.0	19.3	.67	.15	0.75
9	.88	.17	.58	.16	3.2	6.5	.79	.19	0.75
10	.66	.27	.41	.21	10.3	21.1	.58	.24	0.50
11	.76	.26	.46	.21	6.4	14.4	.64	.24	0.75
12	.94	.15	.80	.20	0.8	1.7	.94	.12	1.00

IV. DATA ANALYSIS AND INTERPRETATION

Demonstrating the practicality of predicting module accomplishability requires finding a measure that is both reliable and valid. To avoid redundancy, the factors comprising the measure should be as independent as possible. This analysis addresses these issues in sequence.

A. INTER-RATER RELIABILITY

The reliability of a measure refers to the degree to which the results of measurement are free of error. In this study, the inter-rater reliability of a given factor represents the degree to which the different questionnaire respondents selected the same value for the factor when rating the same module. Table 3 below lists two indicators of relative inter-rater agreement for each factor. The pooled standard deviation is the square root of the mean squared error for all respondents and all modules; it is expressed in the same units as the scale for each factor. Eta^2 is equal to one minus the relative error, where relative error is the error variance divided by the total variance [Ref. 11]. This statistic would equal one if all raters were in complete agreement on every module, and would equal zero if all variance between the ratings for different modules was due solely to differences between raters.

INTER-RATER RELIABILITY

TABLE 3

Factor	pooled s	eta^2
prog	0.175	0.47
est.acc	0.186	0.44
struc	0.195	0.41
size	0.197	0.44
cmplx	0.204	0.46
tool	0.215	0.38
value	0.243	0.13
analy	0.265	0.04
time	88.68	0.09

Interpreting these results, the reliability for the last three factors (Value Judgement, Task Analyzability, and Completion Time) is clearly much lower than for the others; these three are consequently much less useful as predictors. The remaining issue is whether the reliability of the other factors is high enough for them to be utilized in a practical measure of accomplishability, since the indicated reliabilities of 38% to 47% seem low. Low inter-rater reliability can be attributed to either low variability of tasks, or high variability of raters. Reviewing Table 2, low task variability is potentially a problem only for Task Analyzability and Value Judgement. For the remaining factors, rater variability must account for the low reliabilities.

Two points reduce the significance of the low reliability numbers. First, some variation in factor ratings is expected because the factors in the questionnaire represent subjective judgments that are highly dependent on the individual rater's knowledge, experience, and perceptions. For example, the variety of programming tools with which an individual is familiar can greatly influence the value chosen for Tool Availability for a given module, regardless of the tools that actually may exist. As another example, if an individual perceives that a task is poorly structured, then for that individual, the task <u>is</u> poorly structured, even if some other rater may see a well-defined structure in the task. It is also likely that the practical knowledge varied more between the students completing the questionnaire than it would between a group of actual practitioners. The second point is that while the initial Archipelagian Approach paper specified an interval scale for factors to facilitate computation of the Accomplishability Factor, the important issue is the relative ranking of the modules. It is possible for raters to differ on the exact value assigned to each module, yet maintain the same relative ranking. As an illustration, module five in this study clearly has a less structured task to perform than module nine. The Task Structure ratings assigned by the 45 questionnaire respondents varied from 0.00 to 0.75 for module five, and from 0.25 to 1.00 for module nine, which results in relatively low inter-rater reliability. However, 88% of the raters marked module nine as more structured than they marked module five; 9% (four raters) had them even, and only 2% (one person) thought module five was more structured. Considering these points, this researcher believes that the inter-rater reliabilities of Task Programmability, Task Structure, Module Size, Task Complexity, and Tool Availability are not unusually low.

B. VALIDITY

The validity of a measure represents the degree to which it actually measures what it purports to measure. For this study, the coefficient of determination (r^2) of the linear regression of the actual accomplishability values on the evaluation factors provides an external check on the validity of the predictions. Table 4 below lists these coefficients for regressions of both the individual raters' values and the aggregate (mean) values. The factors marked with an asterisk are the ones determined in the previous section to have very low inter-rater reliability.

REGRESSION COEFFICIENTS

TABLE 4

Factor	indiv. r^2	mean r^2
est.acc	35.2%	80.3%
prog	37.0	79.6
struc	32.5	78.7
tool	28.2	74.3
size	32.1	73.7
cmplx	32.0	69.2
*time	6.6	71.8
*value	5.1	39.7
*analy	0.3	6.7

The regression results in Table 4 indicate that while the validity of individual raters' predictions was fairly low, the aggregate results were quite accurate, especially for the overall Estimated Accomplishability judgement. Only for module six, the intentionally "impossible" module, was the aggregate estimated accomplishability greatly different from the actual accomplishability (see Table 2). This was possibly due to a general reluctance for raters to use the low endpoint of the rating scale. Of course, the "actual" values, while based on more information than was available to the questionnaire respondents, still represent a judgement by the researcher and not an absolute standard.

Further examination of Table 4 shows that no single factor has an aggregate validity higher than the summary Estimated Accomplishability. In addition to the simple regressions shown, multiple regressions were performed using combinations of two, three, and four factors. No combination had a coefficient of determination significantly greater than the 80.3% obtained using the raters' Estimated Accomplishability. Since every respondent evaluated all factors for every module, this study cannot determine whether equally accurate results could be obtained by simply asking raters to directly estimate accomplishability without considering other factors, or if consideration of the separate factors contributes to the validity of the accomplishability rating.

The regression results display only minor differentiation between the validity of Task Programmability, Task Structure, Tool Availability, Module Size, and Task Complexity, so the results do not provide a basis for selecting which factors to include in the final accomplishability prediction measure.

C. CORRELATION

The correlation between two variables is a measure of the association between them. For this study, correlation represents the degree to which two factors are measuring the same underlying variable that affects accomplishability. Table 5 on the following page lists the Pearson product moment correlation coefficients for the factors correlated across both raters and tasks.

Inspection of the correlation coefficients indicates that all five factors previously identified as having acceptable inter-rater reliabilities are, in general, highly correlated. This indicates that either the factors are in fact interrelated, or that the correlation is a coincidence caused by chance correlation of the factors in the sample modules in the questionnaire. Intuitively, this researcher feels that Task Structure, Task Programmability, and Task Complexity probably do overlap, but that Tool Availability should be independent. Module Size is probably determined by the accomplishability, not vice-versa. In any event, the correlation coefficients do not provide a basis for selecting which factors to include in the final accomplishability prediction measure any more than did the validity results.

CORRELATION COEFFICIENTS

TABLE 5

	est.acc	struc	prog	cmplx	tool	size	value
struc	0.766						
prog	0.752	0.773					
cmplx	0.729	0.713	0.750				
tool	0.782	0.732	0.704	0.659			
size	0.701	0.619	0.626	0.638	0.570		
value	0.300	0.316	0.292	0.233	0.310	0.199	
analy	0.105	0.029	0.049	0.066	0.046	0.032	0.099

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The analysis of the results from this study supports four main conclusions. (1) The inter-rater reliabilities of the summary judgement Estimated Accomplishability and the factors Task Programmability, Task Structure, Module Size, Task Complexity, and Tool Availability are high enough to allow them to be considered for use as predictors of module accomplishability. The inter-rater reliabilities of Completion Time, Value Judgement, and Task Analyzability are not high enough for them to be considered for use as predictors. (2) While the validity of an individual's accomplishability prediction is not likely to be high, the validity of predictions made using the aggregate results from a group of raters employing the high-reliability factors listed in Conclusion One should be excellent. (3) In light of Conclusions One and Two, this study demonstrates that the basic Archipelagian Approach technique of predicting module implementation feasibility is practical. (4) The high correlations among the high-reliability factors implies that they are interrelated, so it should be possible to utilize fewer than five factors to estimate accomplishability without sacrificing predictive validity. Additional study is required to indicate which factors can be eliminated.

B. RECOMMENDATIONS

This researcher has three recommendations to make as a result of this study. (1) Practitioners who decide to utilize the Archipelagian Approach should employ the aggregate judgement of a group of raters to estimate module accomplishability instead of relying on individual results. (2) The Archipelagian Approach authors should consider revising the AF computation technique to utilize an ordinal scale instead of an interval scale, since relative differences between modules seem to be more important and can probably be estimated more reliably than ratings on a fixed scale. (3) Further research should be conducted on the approach.

C. AREAS FOR FURTHER RESEARCH

Inconclusively answered questions from this study provide topics for additional research. (1) Can accomplishability be estimated in one step, or are multiple evaluation factors as used in this study necessary? (2) Is the correlation observed between the high-reliability factors coincidental, or are the factors really related? An associated question is to determine which of the high-reliability factors can safely be eliminated from the accomplishability prediction measure without sacrificing predictive validity. (3) Would a similar questionnaire completed by a group of Decision Support System development practitioners instead of by Computer Systems Management students result in higher inter-rater reliabilities?

While a great deal of additional research remains before the Archipelagian Approach will be completely validated, this initial study provides some evidence that the concept of predicting the difficulty of implementing proposed modules is practical.

LIST OF REFERENCES

1. Bui, T. X. and Sivasankaran, T. R., Integrating Modular Design with Adaptive Design in DSS Prototyping: An Archipelagian Approach, paper presented at the 20th Hawaii International Conference on System Sciences, January 6-9, 1987.

2. Yourdon, E., Managing the Structured Techniques, Third Edition, pp. 86-91, Yourdon Press, 1986.

3. Page-Jones, M., The Practical Guide to Structured Systems Design, pp. 1-7, Yourdon Press, 1980.

4. Keen, P. W., "Adaptive Design for DSS," Database, v. 12, no. 1-2, pp. 15-25, Fall 1980.

5. Sprague, R. H. and Carlson, E. D., Building Effective Decision Support Systems, pp. 139-141, Prentice-Hall, 1982.

6. Naumann, J. and Jenkins, M., "Prototyping: The New Paradigm for Systems Development," MIS Quarterly, pp. 29-44, September 1982.

7. Horton, D. L., A Decision Support Personnel Monitoring Database System Prototype for the United States Marine Corps, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1985.

8. Geshke, M. J., Bullock, R. A., and Widmaier, L. E., TAC * II: An Expert Knowledge Based System for Tactical Decision Making, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1983.

9. Jones, T. C. and Dolenti, J. E., DSS/MIS Design and Implementation for Lamps Mk III Utilizing a Microcomputer and a Maintenance Personnel Assignment Program, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1985.

10. Keen, P. W. and Morton, M. S., Decision Support Systems: An Organizational Perspective, pp. 126-130, Addison-Wesley, 1978.

11. Nunnally, J. C., Psychometric Theory, pp. 133-136, McGraw Hill, 1967.

APPENDIX

DSS DEVELOPMENT QUESTIONNAIRE

A. Introduction

This questionnaire is part of a research effort to evaluate a new proposal for designing and implementing Decision Support Systems (DSS). The proposal is based on the builder's ability to decompose the proposed system into modules, and estimate in advance how easy each module will be to construct.

The following pages contain narrative descriptions of modules included in proposed DSS, and definitions and rating scales for a set of factors for evaluating them. You will be assigning a score on each factor for each module. Assume that an experienced programming team of average ability will be building the system.

B. Factor Descriptions

1. Task Complexity

a. Definition: The degree to which a task involves a large number of variables, and the intricacy of the interrelationships between variables.

b. Rating Scale:

1.00 - Routine or utility task involving essentially no variables.

0.75 - Simple task involving a few variables and uncomplicated interrelationships.

0.50 - Average task involving a few variables that may have involved or intricate interrelationships, or many variables but simple interdependencies.

0.25 - Complex task involving many variables with intricate interdependencies, some of which are unknown.

0.00 - Virtually insurmountable task involving a very large or infinite number of variables that are elaborately interrelated, with many unknowns.

2. Task Programmability

a. Definition: The degree to which a task can be modeled, or reduced to a step-by-step algorithm.

b. Rating Scale:

1.00 - Trivial task that can easily be performed with a few well-defined, simple steps.

0.75 - Routine task; the problem-solving process may be lengthy or involved, but an algorithm can be developed.

0.50 - Partially non-procedural task for which an algorithm is probably not possible, but which can be modeled essentially completely.

0.25 - Non-procedural task that cannot be completely modeled, but which has some limited aspects that a model can describe.

0.00 - Totally unprogrammable; every aspect of the decision process involved is virtually impossible to model.

3. Task Structure

a. Definition: The degree to which the variables involved in a task and the interrelationships between variables can be identified and precisely defined.

b. Rating Scale:

1.00 - Highly structured; variables and relationships are obvious.

0.75 - Structured; all variables and relationships can be readily defined with limited effort.

0.50 - Partially unstructured; some variables that affect the task are hard to identify, or some inter-relationships are unclear.

0.25 - Mostly unstructured; variables are hard to identify, and interrelationships between variables cannot be precisely defined.

0.00 - Totally unstructured; the variables needed to solve the problem cannot be identified.

4. Task Analyzability

a. Definition: The degree to which analysis of the problem has the potential to identify alternative ways of finding a solution.

b. Rating Scale:

1.00 - Unlimited analysis potential; a correct solution can be reached in a virtually infinite number of ways.

0.75 - High analysis potential; a correct solution could be reached through any of a large number of approaches.

0.50 - Average analysis potential; any of a small number of approaches could lead to a correct solution.

0.25 - Limited analysis potential; only a very limited number of approaches are appropriate for the task.

0.00 - No analysis potential; there is only one correct way to solve the task.

5. Value Judgement

a. Definition: The worth or value of having a module included in the planned system from the user's point of view.

b. Rating Scale:

1.00 - Essential module; the system would be useless if the module was removed.

0.75 - Very desirable module; the usefulness of the system would be severely degraded without the module.

0.50 - Desirable module; the usefulness of the system would be somewhat degraded if the module were deleted.

0.25 - Optional module; the function provided is nice to have, but not necessary for the system to be useful.

0.00 - Worthless module; the user would not even notice if this module were removed from the plans for the system.

6. Tool Availability

a. Definition: The degree to which appropriate hardware, programming languages, models, library software, etc. exist for implementing the proposed module on a computer system.

b. Rating Scale:

1.00 - The module can easily be implemented using any of a variety of available tools.

0.75 - Tools are known to exist; limited research would be necessary to select appropriate ones for the project.

0.50 - Tools probably exist; research will be necessary to identify them. Some known tools would need minor modifications in order to be used for the project.

0.25 - Some tools may exist; however, it is likely that some needed tools will require major modifications or need to be developed from scratch.

0.00 - No tools exist and it is unlikely that any could be developed; the project is not suitable for computer implementation.

7. Module Size

a. Definition: The expected length of the module in lines of code using a typical high-level programming language (BASIC, PASCAL, FORTRAN, etc).

b. Rating Scale:

1.ØØ - Very small; less than 5Ø lines of code.

Ø.75 - Small; 5Ø-1ØØ lines of code.

Ø.5Ø - Medium; 1ØØ-1ØØØ lines of code.

Ø.25 - Large; 1ØØØ-5ØØØ lines of code.

Ø.ØØ - Very large; more than 5ØØØ lines of code.

8. Completion Time

a. Definition: The estimated time that an average programming team would need to complete the detailed design and coding of a module.

b. Rating Scale: Estimated time in man-hours.

9. Estimated Accomplishability

a. Definition: The degree of confidence that a module can be implemented as part of a computer program.

b. Rating Scale:

1.ØØ - Very easy; a routine or trivial module that will require little work.

Ø.75 - Easy; an average programming effort will be required, but few problems are anticipated.

Ø.5Ø - Difficult; probably can be implemented, but some preliminary work is necessary to identify tools or work out a method of attacking the problem.

Ø.25 - Very difficult; the module may not be possible to implement, and a major research effort will be necessary to develop tools before work on the module can even start.

Ø.ØØ - Impossible; the module can not be implemented as a computer program.

C. Module Descriptions

Circle the rating for each factor that, in your judgement, is most appropriate.

1. In a proposed system to assist with making assignments for military officers, this module will accept military ID number as an input, search a data store for selected qualifying information about the officer (pay grade, specialty, etc), then search another data store to compile a list of all upcoming billet assignments that the officer is qualified to fill.

a. Task Complexity	1.0 - .75 - .50 - .25 - 0.0
b. Task Programmability	1.0 - .75 - .50 - .25 - 0.0
c. Task Structure	1.0 - .75 - .50 - .25 - 0.0
d. Task Analyzability	1.0 - .75 - .50 - .25 - 0.0
e. Value Judgement	1.0 - .75 - .50 - .25 - 0.0
f. Tool Availability	1.0 - .75 - .50 - .25 - 0.0
g. Module Size	1.0 - .75 - .50 - .25 - 0.0
h. Completion Time	(Estimated man-hours) _____
i. Accomplishability	1.0 - .75 - .50 - .25 - 0.0

2. In a proposed system to assist the Tactical Action Officer (TAO) on a surface ship in making tactical decisions, this module will contain routines to allow

the user to input contact report data as it is received from the ship's sensors.

(NOTE: In the questionnaires used for the survey, a rating scale like the one on the previous page was provided for each module to allow responses to be recorded. In the interest of brevity, the rest of the scales are not reproduced in this Appendix.)

3. In the same TAO system, this module will take contact report data from the input module and maintain a contact database, dead-reckoning as necessary to maintain updated positions on all contacts between reports.

4. In the TAO system, this module will correlate information from the different sensors, and attempt to classify and identify contacts based on reported characteristics. It will also provide an ad-hoc query capability into the contact database.

5. In the TAO system, this module will search through a knowledge base of tactical directives and policies for required actions in the current tactical situation, and search through a knowledge base of stored historical conflicts for similar situations. If a match is found, the successful tactics used in the historical situation will be modified as appropriate to fit the current situation and presented to the operator.

6. In the TAO system, this module will analyze the tactical situation independently of the historical knowledge-based module and work out the optimum tactics for the ship to follow.

7. In a system intended to assist managers with financial planning, this module will use a flexible, English-like dialog to allow users to specify the criteria for evaluating alternatives (net present value, return on sales, profit margin, payback period, etc) and select the type of financial problem to solve (merger/acquisition, project analysis, forecasting, or cash-flow analysis).

8. In the financial planning system, this module will use computational routines such as goal programming, exponential smoothing, or linear regression to solve financial problems. The appropriate technique is selected automatically, depending on the decision criteria and type of problem previously selected by the user.

9. In a proposed system designed to optimize the assignment of personnel to remote-site work teams for temporary projects, this module contains routines to input, update, and edit data about the available personnel and the job positions to be filled.

1Ø. In the personnel assignment system, this module will determine the "payoff" or value of assigning each worker to each possible position. Since selection for a work team means family separation and difficult living conditions for the duration of the project, the goal is to spread the assignments as evenly as possible over all available qualified personnel.

11. In the personnel assignment system, this module will compute the optimum solution for the payoff matrix utilizing the assignment method of linear programming.

12. In the personnel assignment system, this module will print a report listing the optimum assignments.

IV. Computer Background Information

Since estimating how difficult a programming project will be is a subjective activity that varies with the background and experience of the person making the estimation, the following information is requested to allow your responses to be grouped with others that have a similar background. Please circle or fill in the appropriate choices.

1. Curriculum: 367 Other (specify) ____________

2. Quarter: 5th Other (specify) ____________

3. Have you taken the following courses at NPS?

a.	Structured Programming in Pascal	YES	NO
c.	Software Economics	YES	NO
b.	Software Design	YES	NO
d.	Decision Support Systems	YES	NO
e.	Artificial Intelligence	YES	NO

4. Do you have any experience in software design or development other than as a result of NPS classes?

YES NO

If YES, state how many months, and briefly explain the nature of the work:

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. T.R. Sivasankaran, Code 54SJ 5
   Dept. of Administrative Science
   Naval Postgraduate School
   Monterey, California 93943-5000

4. LT Stephen G. Albers 2
   1251 Fuhrman Road
   Cincinnati, Ohio 45215

Albers
The Archipelagian Approach to DSS prototyping: an empirical study.

Thesis
A3365
c.1

Albers
The Archipelagian Approach to DSS prototyping: an empirical study.